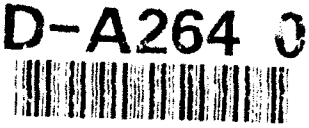
REPORT DOCUMENTATI

AD-A264 016



pproved
o 0704 0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1993	Final 15 Aug 92-14 Feb 93

4. TITLE AND SUBTITLE
Algorithms for Point Set Congruence

5. FUNDING NUMBERS
DAAL03-92-G-0378

6. AUTHOR(S)
Paul J. Heffernan

DTIC
ELECTE
MAY 1 0 1993
S C D

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Memphas State University
Memphis, TN 38152

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING MONITORING AGENCY REPORT NUMBER
ARO 30778.3-MA

11. SUPPLEMENTARY NOTES
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The primary purpose of the project supported by this grant was to expand research in point matching, by applying the perturbation technique and the approximate algorithm paradigm to more complex problem formulations. During the tenure of the grant the investigator has successfully worked towards this goal. The main results of this research are contained in the paper "Generalized approximate algorithms for point set congruence". This paper is still undergoing revisions, and a copy of the most recent draft is included in this report. The paper introduces a device called the (ϵ, k)-map, which is a more general measure of point set congruence than the one previously

Cont'd on reverse side

14. SUBJECT TERMS	15. NUMBER OF PAGES
Algorithms, Point Set Congruence, Pergurbation Technique	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

examined by the investigator. The paper discusses ways to construct the (ε, k)-map. While the investigator's previous work was limited to equal cardinality, planar point sets, this current work has expanded attention both to point sets of unequal cardinality and to point sets in higher dimensions. The investigator has even begun the study of projective congruence, in which one compares a 2-dimensional image with the family of projections of a 3-dimensional model.

Algorithms for Point Set Congruence

Final Report

Principal Investigator: Paul J. Heffernan
Memphis State University
Memphis, Tennessee 38152

March 31, 1993

Grant No. DAAL03-92-G-0378

Approved for public release;
distribution unlimited

Table of Contents

Final Report: Algorithms for Point Set Congruence
ARO Grant No. DAAL03-92-G-0378

Under the support of this grant, the investigator studied the question of congruence between two sets of points. The problem draws its motivation from computer vision. One point set can be thought of as a model, and the other as an image. A given model could represent a symbol, such as a letter of the alphabet, or a landscape or other environment. An image is that which is seen by an observer. We wish to know if an observed image matches a certain model.

Before the tenure of this grant began, this investigator had begun research into several formulations of this problem. The investigator had improved upon results of other researchers by introducing a technique that perturbs the input points slightly in order to exploit the resulting geometric structure of the point sets. The investigator had applied this technique to the case of equal cardinality, planar point sets. The resulting algorithms were called "approximate algorithms," because they traded some precision for faster runtimes.

The primary purpose of the project supported by this grant was to expand research in point matching, by applying the perturbation technique and the approximate algorithm paradigm to more complex problem formulations. During the tenure of the grant the investigator has successfully worked towards this goal. The main results of this research are contained in the paper "Generalized approximate algorithms for point set congruence". This paper is still undergoing revisions, and a copy of the most recent draft is included in this report. The paper introduces a device called the (ε, k)-map, which is a more general measure of point set congruence than the one previously examined by the investigator. The paper discusses ways to construct the (ε, k)-map. While the investigator's previous work was limited to equal cardinality, planar point sets, this current work has expanded attention both to point sets of unequal cardinality and to point sets in higher dimensions. The investigator has even begun the study of projective congruence, in which one compares a 2-dimensional image with the family of projections of a 3-dimensional model.

1 Equal cardinality, planar point sets

The case of equal cardinality, planar point sets was the main focus of this investigator before the tenure of this grant. The major results of this work are contained in the paper "Approximate decision algorithms for point set congruence," written by this investigator with Stefan Schirra. Final revisions on the paper were made by the authors during the period of the grant, and the paper has been accepted in revised form by the journal *Computational Geometry: Theory and Applications*. The content of this paper is important to the main body of work pursued under the grant, since the recent work has studied generalizations of the problem formulated in this paper. We summarize below the problem studied in "Approximate decision algorithms for point set congruence," and the approximate algorithm paradigm.

The paper studies the problem of *ε-congruence*. Assume that we are given two planar point sets A and B of equal cardinality, n, and a tolerance value ε. An isometry is a mapping in $\mathbb{R}^2$ that preserves distances. We say that A and B are *ε-congruent* if there exists an isometry I and a bijection $\ell : A \rightarrow B$ such that $dist(\ell(a), I(a)) \leq \varepsilon$, for all $a \in A$, where $dist(\cdot, \cdot)$ is the distance function of our chosen metric (typically L_2). Intuitively, we can approach this problem by thinking of each point set being put on a piece of paper; the sets are ε-congruent if it is possible to slide one piece of paper over the other and match the points in such a way that the points of each pair are within distance ε of each other. Important special cases of this formulation are obtained by restricting the isometry to be a translation, T, or a rotation around a fixed point d, R_d.

Determining if point sets A and B are ε-congruent for a given ε is referred to as the decision problem. The optimization problem asks for $\varepsilon_{opt}(A, B)$, the smallest value ε for which A and B are ε-congruent. It is possible to use any algorithm for the decision problem to approximate $\varepsilon_{opt}(A, B)$, by a search procedure.

Early algorithms for the ε-congruence decision suffered problem from large run-times. A concept introduced by Stefan Schirra to obtain more efficient algorithms is the *approximate decision algorithm*. A decision algorithm is *(α, β)-approximate*, if, for any $\varepsilon \notin [\varepsilon_{opt}(A, B) - \alpha, \varepsilon_{opt}(A, B) + \beta]$, it correctly answers a query, and for $\varepsilon \in [\varepsilon_{opt}(A, B) - \alpha, \varepsilon_{opt}(A, B) + \beta]$, it either answers correctly or chooses not to answer. As we intuitively believe that it is more difficult to test for ε-congruence if ε is close to $\varepsilon_{opt}(A, B)$, we

would be willing to accept an algorithm that correctly answers queries for values of ε not near $\varepsilon_{opt}(A, B)$, but sometimes chooses not to answer when ε is near $\varepsilon_{opt}(A, B)$, if that algorithm provides substantial time savings. An (α, β)-approximate algorithm has the desirable property that it will not return an incorrect answer; if it is not sure, it will simply say that it does not know the answer. For this reason, the designation "approximate" is perhaps a misnomer; an alternate title for an algorithm that sometimes chooses not to answer is "incomplete decision algorithm" (as opposed to a "complete decision algorithm," which answers all queries).

2 Generalized approximate algorithms

While "Approximate decision algorithms for point set congruence" develops approximate algorithms for the original formulation of ε-congruency, this formulation is unnecessarily constrictive, which motivates the desire to generalize results. This generalization has been the main activity of the project, and the principal results are presented in the paper, "Generalized approximate algorithms for point set congruence." While this paper is attached to this report, we will summarize briefly its content.

First of all, the work of this project has dropped an assumption of the earlier work: the point sets are not required to have equal cardinality. This greatly enlarges its applications. Secondly, the work examines point sets not only in 2-dimensional space but in 3-dimensions also; it even begins an examination of comparing a 2-dimensional set to a 3-dimensional set.

The present work also introduces a stronger measure for congruency. The original decision problem merely tells whether it is possible to match the points such that each point is within the tolerance distance; the present work asks for the maximum number of pairs that can be placed within the tolerance distance.

This notion is formalized as follows. We are given two point sets A and B in $\mathbb{R}^d$, with $|A| = n$, $|B| = m$, $n \leq m$, a family of isometries $\mathcal{I}$, and a metric $dist(\cdot, \cdot)$ on $\mathbb{R}^d$. We define two functions,

$$k_{max} : \mathbb{R}^+ \rightarrow \{2, \ldots, n\} \quad \text{and} \quad \varepsilon_{opt} : \{2, \ldots, n\} \rightarrow \mathbb{R}^+,$$

which describe the congruence between A and B. The definitions are as follows.

- We say that A and B are (ε,k)-*congruent* if there exist $I \in \mathcal{I}$ and an injection (1-to-1 function) $\ell : A \rightarrow B$ such that $dist(\ell(a), I(a)) \leq \varepsilon$ for at least k points $a \in A$.

- For $\varepsilon > 0$, $k_{max}(\varepsilon)$ (or $k_{max}(\varepsilon, A, B)$) is the maximum value k such that A and B are (ε,k)-congruent.

- For $k \in \{2, \ldots, n\}$, $\varepsilon_{opt}(k)$ (or $\varepsilon_{opt}(k, A, B)$) is the smallest value $\varepsilon > 0$ such that A and B are (ε,k)-congruent.

Combining the notions behind the functions k_{max} and ε_{opt} leads naturally to a structure called the (ε,k)-*map*, which summarizes the congruency relationship between two point sets.

The investigator has generalized the definition of an approximate decision algorithm to correspond to the generalized definition of approximate congruence. The result is a powerful framework for studying congruence of point sets. The investigator has developed some algorithms for the various formulations of the generalized point matching question, and these are given in the attached paper. The investigator hopes to continue to improve the algorithms for this problem.

3 Other work

While the work of this project has been primarily concerned with point set matching, the investigator has benefitted from the support of the grant while pursuing research on some related topics. Point matching is a topic in computational geometry, since it asks for algorithmic solutions to geometrical problems. The investigator has recently worked on some other computational geometry topics as well.

The investigator has studied a question in polygonal visibility known as the *two-guard* problem. Given a simple polygon P with vertices s and t, the *straight walk* problem asks whether we can move two points monotonically on P from s to t, one clockwise and one counterclockwise, such that the points are always co-visible. In the *counter walk* problem, both points move clockwise, one from s to t and the other from t to s. Optimal $\Theta(n)$ constructive algorithms for both problems are found. The results are obtained by examining the structure of the restrictions placed on the motion of the

two points, and by employing properties of shortest paths and shortest path trees. The results are described in the paper "An optimal algorithm for the two-guard problem," which has been submitted to the journal *International Journal on Computational Geometry*; copies of the paper were furnished to the A.R.O. at the time of the paper's submission to the journal. This paper will be presented at the 9th ACM Symp. on Computational Geometry, May 1993.

The investigator has researched the topic of *t-spanners*. Let V be a set of n points in 3-dimensional Euclidean space. A subgraph of the complete Euclidean graph is a t-spanner if for any u and v in V, the length of the shortest path from u to v in the spanner is at most t times $d(u, v)$. The investigator, with Gautam Das and Giri Narasimhan, has shown that for any $t > 1$, a greedy algorithm produces a t-spanner with $O(n)$ edges, and total edge weight $O(1) \cdot wt(MST)$, where MST is a minimum spanning tree of V. This result is described in "Optimally sparse spanners in 3-dimensional Euclidean space," which will be presented at the 9th ACM Symp. on Computational Geometry, May 1993.

The investigator recently made final revisions on "Linear-time algorithms for weakly-monotone polygons," which has been accepted in revised form by the journal *Computational Geometry: Theory and Applications*. The paper introduces a new class of simple polygons called the weakly-monotone class. It provides a linear-time algorithm that determines whether a simple polygon is weakly-monotone, and a brief, linear-time algorithm that triangulates a weakly-monotone polygon.

4 Papers

"Generalized approximate algorithms for point set congruence," manuscript, 1993.

"Approximate decision algorithms for point set congruence," in *Proc. of the 8th ACM Symposium on Computational Geometry*, 1992; to appear in *Computational Geometry: Theory and Applications.*

"An optimal algorithm for the two-guard problem," to be presented at *9th ACM Symposium on Computational Geometry*, San Diego, 1993.

"Optimally sparse spanners in 3-dimensional Euclidean space," to be presented at *9th ACM Symposium on Computational Geometry*, San Diego, 1993. Co-Authors: G. Das and G. Narasimhan.

"Linear-time algorithms for weakly-monotone polygons," to appear in *Computational Geometry: Theory and Applications.*

Generalized Approximate Algorithms for Point Set Congruence

Paul J. Heffernan *

Dept. of Mathematical Sciences, Memphis State University, Memphis, TN 38152

Abstract

We address the question of determining if two point sets are congruent. This task consists of defining the concept of point congruence, and developing algorithms that test for it. We introduce the (ε,k)-*map*, a device which pictorially represents the degree of congruence between two point sets. Point set congruence has been studied by previous researchers, but we feel that our definitions offer a more general and powerful approach to the problem. By using the paradigm of *approximate algorithms*, we are able to construct the (ε,k)-map efficiently.

1 Introduction

In this paper we address the following question: given two point sets in $\mathbb{R}^d$, what do we mean when we say that they are "congruent," and how do we determine congruency? To answer this question, we must define congruency, and provide efficient algorithms that test for it. The practical motivation behind our study comes from computer vision, where an observed image is compared to a hypothesized model. While previous researchers have formalized and studied this problem, we feel that the formulations often have been too restrictive. In particular, an elegant formulation has been introduced by [AMWW] (also [Ba]) and studied further by [HS]. It is this formulation, which we call ε-congruence, which we generalize here, in an attempt to lessen its restrictions while preserving its strengths.

Roughly speaking, the ε-congruence formulation is as follows. The input is assumed to be two equal cardinality, planar point sets, with a given metric (usually L_2 or L_∞) and a given family of isometries (an isometry is a mapping from $\mathbb{R}^d$ to itself that preserves distances). There are two problem versions, decision and optimization. In the decision problem, one asks for a given value $\varepsilon > 0$ whether an isometry from the family can be applied to one point set and a matching found between the sets such that the points in each matched pair are within distance ε of each other. In the optimization problem, one asks for the smallest value ε that returns an affirmative answer in the decision problem.

*Supported in part by the U.S. Army Research Office, Grant No. DAAL03-92-G-0378

Formulation of the model as a point set is appropriate in applications where the model consists of a number of small pieces, such as locations on a map, or the model is a figure that can be represented by its key boundary points, such as letters of an alphabet. The ε-congruence formulation has both advantages and drawbacks. One of its main strengths is that is allows for the real-world possibility of noisy input—that is, imprecisions in recording the image points—by considering a pair of points to be matched if they are merely within a tolerance distance ε of each other. Every point of both the model and image is important, since each point must be matched to a nearby partner. This allows us to distinguish between models that vary in only a few points. On the other hand, if even one image point encounters an amount of noise exceeding the tolerance value, we may falsely answer that an image does not resemble a model. Another concern with the strict definition of [AMWW] is the requirement that the two point sets have equal cardinality. Among the imperfections of image registrations are missing points, which fail to appear on the image, and spurious points, which are erroneously introduced into the image. The possibility of such events renders the equal cardinality assumption impractical. Additionally, in [AMWW] attention is limited to 2-dimensional point sets.

Our goal in this paper is to abandon the rigidity of the traditional definition of ε-congruency. In lieu of the decision problem, we propose the following generalized version: for a given $\varepsilon > 0$, find the largest number of point pairs that can be placed within distance ε of each other, under some matching and isometry from the family. The generalized optimization problem asks, for a given value k, to find the smallest ε for which the decision problem returns a value of at least k. We introduce and show how to compute a structure called the (ε,k)-*map*, which stores the complete generalized solution. We drop the assumption that the point sets must be of equal cardinality, and we expand our methods to higher dimensional point sets. We consider a variety of metrics and families of isometries. We even consider the case of *projection congruency*, in which a 2-dimensional image is compared to a 3-dimensional model. As a result of these efforts, we build a more powerful framework for congruency of point sets.

While Alt, et al. [AMWW] give algorithms for the ε-congruence problem, most suffer from high worst-case run-times. A way to obtain efficient algorithms while preserving the formulation is through the use of *approximate algorithms* for ε-congruence, first introduced by Schirra [Sc1, Sc2]. Faster approximate algorithms have subsequently been developed by Heffernan and Schirra [HS]. In order to develop algorithms for our generalized definition of point set congruence, we draw on the techniques of [HS], and develop a body of approximate algorithms.

2 Generalized approximate congruence

In the point set congruence problem, we are given sets A and B. We wish to match points of A to points of B so that as many pairs as possible are as close as possible. In this sense we have a bicriteria problem with a trade-off between the two goals. In order to obtain workable formulations, we can fix one goal while optimizing the other. Under this approach,

one problem consists of matching the points in order to maximize the number of "close" pairs, where "close" is determined by a fixed tolerance value. The corresponding problem fixes the number of close pairs and asks for the minimum possible closeness parameter. In either problem, we are allowed to impose an isometry from a given family on one of the point sets.

We formalize these notions as follows. We are given two point sets A and B in $\mathbb{R}^d$, with $|A| = n$, $|B| = m$, $n \leq m$, a family of isometries $\mathcal{I}$, and a metric $dist(\cdot,\cdot)$ on $\mathbb{R}^d$. We define two functions,

$$k_{max} : \mathbb{R}^+ \rightarrow \{2,\ldots,n\} \text{ and } \varepsilon_{opt} : \{2,\ldots,n\} \rightarrow \mathbb{R}^+,$$

which describe the congruence between A and B. The definitions are as follows.

- We say that A and B are (ε,k)-*congruent* if there exist $I \in \mathcal{I}$ and an injection (1-to-1 function) $\ell : A \rightarrow B$ such that $dist(\ell(a), I(a)) \leq \varepsilon$ for at least k points $a \in A$.
- For $\varepsilon > 0$, $k_{max}(\varepsilon)$ (or $k_{max}(\varepsilon, A, B)$) is the maximum value k such that A and B are (ε,k)-congruent.
- For $k \in \{2,\ldots,n\}$, $\varepsilon_{opt}(k)$ (or $\varepsilon_{opt}(k, A, B)$) is the smallest value $\varepsilon > 0$ such that A and B are (ε,k)-congruent.

Combining the notions behind the functions k_{max} and ε_{opt} leads naturally to a structure that we call the (ε,k)-*map*, an example of which is pictured in Figure 1. The (ε,k)-map plots k against ε, and it consists of both a YES region and a NO region of ordered pairs (ε,k), such that A and B are (ε,k)-congruentif and only if (ε,k) is in the YES region. The (ε,k)-map allows one to see the interaction between the two goals of obtaining a small closeness parameter and a large matching. One wishes for a point in the upper-left corner of the map, while the YES region is anchored in the lower-right corner. The boundary between the two regions is a step-wise curve from lower-left to upper-right (the peculiarity of a discontinuous curve serving as a boundary occurs because k assumes discrete values). In essense, the (ε,k)-map is a picture of the congruence between A and B.

The (ε,k)-map offers some clear advantages over the traditional definition of ε-congruence. The latter deals only with matchings of size n, and therefore limits its perspective to the top line of the (ε,k)-map. Much additional information is contained in the full (ε,k)-map: two pairs of point sets may have the same value of $\varepsilon_{opt}(n)$ but a sharp difference at $\varepsilon_{opt}(n-i)$ for a small integer i. Indeed, large recording errors for a few image points can increase $\varepsilon_{opt}(n)$ greatly, while $\varepsilon_{opt}(n-i)$, where i equals the number of such "outliers," gives a much truer picture. We should also note that the (ε,k)-map drops the assumption of equal cardinality point sets. The relaxation allows the formulation to encompass the case of lost or spurious points in the image. Once such points exist, the idea of a "perfect matching" between the two point sets becomes less meaningful, and the concept of the (ε,k)-map more appropriate. For these reasons we feel that the (ε,k)-map is the true manner in which to think about approximate congruence.

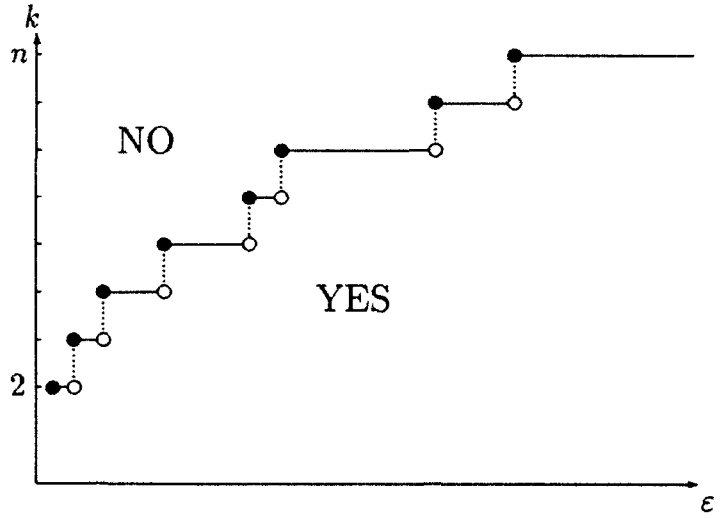


Figure 1: The (ε,k)-map

3 Generalized approximate algorithms

Approximate decision algorithms were introduced in [Sc1, Sc2], and expanded in [HS]. Here we present a generalized description of approximate algorithms suitable for (ε,k)-congruence. For fixed ε and k, the decision problem asks whether input point sets A and B are (ε,k)-congruent. It is natural to believe that this question is most difficult when ε is near $\varepsilon_{opt}(k)$, so we will allow an approximate algorithm not to answer in such a situation. An (α, β)-*approximate decision algorithm* for (ε,k)-congruence of A and B is one which (1) correctly determines whether A and B are (ε,k)-congruent whenever $\varepsilon \notin [\varepsilon_{opt}(k) - \alpha, \varepsilon_{opt}(k) + \beta]$, and (2) either correctly determines (ε,k)-congruence or chooses not to answer when $\varepsilon \in [\varepsilon_{opt}(k) - \alpha, \varepsilon_{opt}(k) + \beta]$. We call $[\varepsilon_{opt}(k) - \alpha, \varepsilon_{opt}(k) + \beta]$ the *imprecision interval.* A major strength of approximate decision algorithms is that they never return an incorrect answer; if a particular query (ε,k) is too difficult, the algorithm simply says that it is unsure. Previous approximate decision algorithms have been defined only for $k = n$.

Earlier, we stated two optimization problems: maximizing k for fixed ε (i.e. computing $k_{max}(\varepsilon)$) and minimizing ε for fixed k (computing $\varepsilon_{opt}(k)$). We wish to extend the concept of an approximate algorithm to these optimization problems.

Let us first consider $k_{max}(\varepsilon)$. For a fixed ε, the ordered pair (ε,k) is an instance of the decision problem, for $k = 2, \ldots, n$. For all $k \leq k_{max}(\varepsilon)$, A and B are (ε,k)-congruent, while for all $k > k_{max}(\varepsilon)$ they are not. Therefore, a (complete) algorithm for the k_{max}-optimization problem partitions the set $\{2, \ldots, n\}$ of values for k into two sets: $\{2, \ldots, k_{max}(\varepsilon)\}$, which we call the YES set, and $\{k_{max}(\varepsilon) + 1, \ldots, n\}$, the NO set (one of these sets may be empty).

An approximate algorithm for the $k_{max}(\varepsilon)$ problem partitions $\{2, \ldots, n\}$ into three sets, $\{2, \ldots, k_1\}$, $\{k_1 + 1, \ldots, k_2\}$, and $\{k_2 + 1, \ldots, n\}$, labelled YES, MAYBE, and NO, respectively, where $k_1 \leq k_{max}(\varepsilon) \leq k_2$ (one or two of these sets may be empty). For values of k in the MAYBE set, the algorithm is unable to determine whether $k \leq k_{max}(\varepsilon)$. We call such

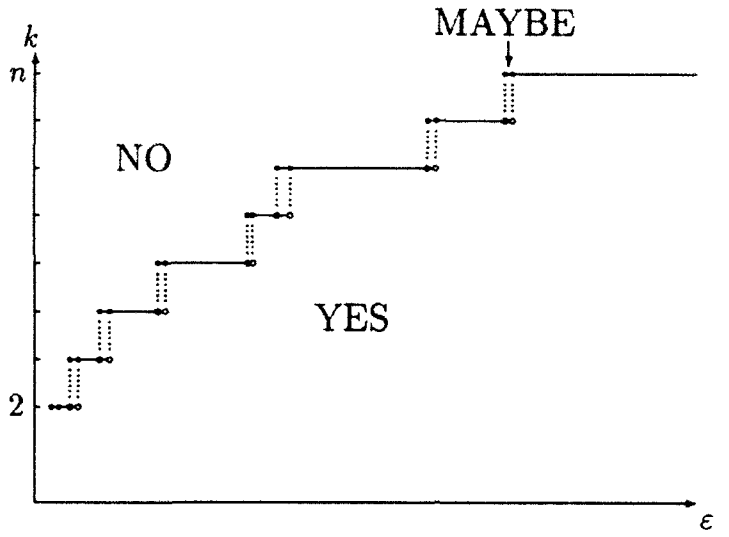


Figure 2: The (ε,k,ν)-approximate map

an algorithm (α,β)-approximate if, for every k in the MAYBE set, ε is in the corresponding imprecision interval, i.e. $\varepsilon \in [\varepsilon_{opt}(k) - \alpha, \varepsilon_{opt}(k) + \beta]$.

Now let us consider the situation of a fixed value k. The interval $\mathbb{R}^+$ of possible values for ε is divided into the NO subinterval $(0,\varepsilon_{opt}(k))$ and the YES subinterval $[\varepsilon_{opt}(k),\infty)$; these intervals are returned by a (complete) algorithm for the $\varepsilon_{opt}(k)$ problem when it is given input k. An approximate algorithm for the $\varepsilon_{opt}(k)$ problem, when given k, returns intervals $(0,\varepsilon_1)$, $[\varepsilon_1,\varepsilon_2)$, and $[\varepsilon_2,\infty)$, labelled NO, MAYBE, and YES, respectively, where $\varepsilon_1 \leq \varepsilon_{opt}(k) \leq \varepsilon_2$. As above, the MAYBE set represents those values of ε for which the algorithm is not able to determine (ε,k)-congruence. Such an algorithm is (α,β)-approximate if, for every ε in the MAYBE set, ε is in the corresponding imprecision interval; that is, $\varepsilon_1, \varepsilon_2 \in [\varepsilon_{opt}(k) - \alpha, \varepsilon_{opt}(k) + \beta]$.

4 Building the (ε,k,ν)-approximate map

In this section we show how to use a (γ,γ)-approximate algorithm for the $k_{max}(\varepsilon)$ problem in order to build a (ε,k,ν)-*approximate map* (see Figure 2). A (ε,k,ν)-approximate map is similar to an (ε,k)-map, except that in the approximate map, $\varepsilon_{opt}(k)$ for a given k is represented, not as a single point, but as an interval of length no more than ν that contains $\varepsilon_{opt}(k)$. The (ε,k)-congruent region is separated from the non-congruent region not by a step-wise curve, but by a region, which we call the imprecision region (or, the MAYBE region).

Our approach consists of a two-part search. The first part, which we call Phase I, finds for each $k \in \{2,\ldots,n\}$ an interval $(c,2c)$ that contains $\varepsilon_{opt}(k)$. In Phase II, a binary search is performed on each interval $(c,2c)$ until the interval size is at most ν.

Since a (γ,γ)-approximate algorithm is more precise for small values of γ, we expect its run-time to vary inversely with γ. Actually, we will see that the run-times for our approximate algorithms for the $k_{max}(\varepsilon)$ problem vary directly with ε/γ, so we attempt to keep this value small when constructing our (ε,k)-map algorithm. We will see that the time-complexity of Phase II dominates that of Phase I. We will describe the general (ε,k)-map estimation method now. Later we will give (γ,γ)-approximate algorithms for the $k_{max}(\varepsilon)$ problem in various dimensions $\mathbb{R}^d$ and under various families of isometries $\mathcal{I}$, and after analyzing each such algorithm we will compute the time-complexity of the corresponding (ε,k)-map estimation.

The idea behind Phase I is simple: compute $k_{max}(\varepsilon)$, initially for a small value ε, then repeatedly double ε and recompute $k_{max}(\varepsilon)$, until ε is as large as $\varepsilon_{opt}(k)$. At each step we use a (γ,γ)-approximate algorithm, where we set $\gamma = \varepsilon/2$. Let us first describe how Phase I works for a single value k, and then generalize the approach for all k.

We begin by setting $\varepsilon = \nu$, and then alternating the steps of testing for (ε,k)-congruence by using a $(\varepsilon/2,\varepsilon/2)$-approximate algorithm, and doubling ε, until a test returns an answer of YES or MAYBE. (Note that an answer of YES (NO) to an (ε,k)-congruence test implies that $\varepsilon \geq \varepsilon_{opt}(k)$ ($\varepsilon < \varepsilon_{opt}(k)$).) At this point, we suspect that $\varepsilon_{opt}(k)$ is roughly within a factor of 2 of ε, since the test for $(\varepsilon/2,k)$-congruence returned a NO answer while the one for (ε,k)-congruence returned YES or MAYBE. If the answer for (ε,k)-congruence is YES, then we realize that $\varepsilon_{opt}(k) \in (\varepsilon/2,\varepsilon)$, which is an interval of the form $(c,2c)$. If the answer is MAYBE, we see that $\varepsilon_{opt}(k) \in (\varepsilon/2,3\varepsilon/2)$, since we used a $(\varepsilon/2,\varepsilon/2)$-approximate algorithm to test for (ε,k)-congruence. This interval is not of the form $(c,2c)$, but we can trim it to this form through one additional test for (ε,k)-congruence with $\gamma = \varepsilon/4$: an answer of YES, MAYBE, or NO, respectively, yields an interval $(\varepsilon/2,\varepsilon)$, $(3\varepsilon/4,5\varepsilon/4)$, or $(\varepsilon,3\varepsilon/2)$.

There is one special case, obtaining a YES or MAYBE answer on the initial test, when $\varepsilon = \nu$. Either answer implies that we are immediately finished not only with Phase I, but with Phase II as well. An answer of YES implies $\varepsilon_{opt}(k) \in (0,\nu)$, while MAYBE implies $\varepsilon_{opt}(k) \in (\nu/2,3\nu/2)$; thus, in either case we have found an interval of size ν that contains $\varepsilon_{opt}(k)$.

If we use a (γ,γ)-approximate algorithm for the $k_{max}(\varepsilon)$ problem, then we can perform Phase I simultaneously for all values $k = 1,\ldots,n$. The (γ,γ)-approximate algorithm returns an answer of NO, MAYBE, or YES for each k, so we alternate performing this algorithm with doubling ε until all values of k have received a YES or MAYBE answer ($k = n$ will be the last to receive one of these answers).

Phase II is quite simply a binary search. Beginning with $\varepsilon_{opt}(k) \in (c,2c)$, we proceed to narrow the interval to width ν. A step begins with $\varepsilon_{opt}(k) \in (\ell,u)$. We test for (ε,k)-congruence, where $\varepsilon = (\ell+u)/2$, and $\gamma = (u-\ell)/4$. With a YES answer we set $u \leftarrow (\ell+u)/2$ and repeat, with NO we set $\ell \leftarrow (\ell+u)/2$ and repeat, and with MAYBE, our choice of γ allows us to set $\ell \leftarrow (3\ell+u)/4$, $u \leftarrow (\ell+3u)/4$, and repeat.

5 Algorithms for $k_{max}(\varepsilon)$

In this section we describe (γ,γ)-approximate algorithms for $k_{max}(\varepsilon)$, under various isometry families. We will use the Euclidean metric, although our algorithms work for any L_p metric. Our algorithms are based largely on the (γ,γ)-approximate decision algorithms for ε-congruence of [HS].

All approximate algorithms employ a similar strategy; they (1) discretize the set of isometries, $\mathcal{I}$, into a finite set of candidates, $\mathcal{C}$, and (2) test the point sets for congruence under each candidate. The methods of [HS] employ an additional tactic of perturbing each point in A and B in order to impose structure; the resulting multisets are denoted $A^{\#}$ and $B^{\#}$. This method consists of laying a lattice onto $\mathbb{R}^d$, with adjacent lattice points distance λ apart, where λ is proportional to γ. $A^{\#}$ and $B^{\#}$ are obtained by moving each point of A and B to the nearest lattice point. Note that no point is perturbed more than $\sqrt{d}\lambda/2$. We will see later how working with $A^{\#}$ and $B^{\#}$ instead of A and B allows an improvement in time-complexity.

Each algorithm requires that the following conditions hold for some fixed γ_1, γ_2.

Condition 1: For each $I \in \mathcal{I}$ that admits (ε,k)-congruence for some $k \in \{2,\ldots,n\}$, there exists a candidate isometry $C \in \mathcal{C}$ such that $dist(C(a^{\#}), I(a^{\#})) \leq \gamma_1$ for all $a^{\#} \in A^{\#}$.

Condition 2: For each $k \leq n$, if $A^{\#}$ and $B^{\#}$ are $(\varepsilon - \gamma_2, k)$-congruent for a candidate isometry $C \in \mathcal{C}$, then A and B are (ε,k)-congruent.

Condition 3: For each $k \leq n$, if $A^{\#}$ and $B^{\#}$ are $(\varepsilon + \gamma_1 + \gamma_2, k)$-congruent for *no* candidate isometry $C \in \mathcal{C}$, then A and B are not (ε,k)-congruent.

In the discussion that follows, we let $k_{max}(\mu, A^{\#}, B^{\#}; C)$ represent the size of the largest matching between $A^{\#}$ and $B^{\#}$ with tolerance μ under isometry C. Table 1 gives an approximate algorithm for $k_{max}(\varepsilon, A, B)$, assuming we have constructed $\mathcal{C}$ and have a procedure that computes $k_{max}(\mu, A^{\#}, B^{\#}; C)$.

[1] Compute set of candidates $\mathcal{C}$
[2] $k_{YES} \leftarrow 1$, $k_{MAYBE} \leftarrow 1$
[3] **for all** $C \in \mathcal{C}$ **do**
[4] **if** $k_{max}(\varepsilon + \gamma_1 + \gamma_2, A^{\#}, B^{\#}; C) > k_{MAYBE}$
then $k_{MAYBE} \leftarrow k_{max}(\varepsilon + \gamma_1 + \gamma_2, A^{\#}, B^{\#}; C)$ **fi**
[5] **if** $k_{max}(\varepsilon - \gamma_2, A^{\#}, B^{\#}; C) > k_{YES}$
then $k_{YES} \leftarrow k_{max}(\varepsilon - \gamma_2, A^{\#}, B^{\#}; C)$ **fi od**;
[6] YES $\leftarrow \{2,\ldots,k_{YES}\}$
[7] MAYBE $\leftarrow \{k_{YES}+1,\ldots,k_{MAYBE}\}$
[8] NO $\leftarrow \{k_{MAYBE}+1,\ldots,n\}$

Table 1: (γ,γ)-approximate algorithm for $k_{max}(\varepsilon, A, B)$

Theorem 1 *If Conditions 2 and 3 are satisfied, then the algorithm of Table 1 is a (γ, γ)-approximate algorithm for $k_{max}(\varepsilon, A, B)$, where $\gamma = \gamma_1 + 2\gamma_2$.*

In our algorithms, $\gamma_2 = 2\sqrt{d}\lambda/2$, since no point of A or B is perturbed more than $\sqrt{d}\lambda/2$. We will also choose γ_1 proportional to λ for each algorithm, so $\gamma = \Theta(\lambda)$.

Implementation of an algorithm of this form requires (1) a method for constructing $\mathcal{C}$, and (2) a method for computing $k_{max}(\mu, A^\#, B^\#; C)$. The first question depends on the family of isometries $\mathcal{I}$, and will be explored below. The second question leads us a graph theoretic formulation and solution.

The matching problem between $A^\#$ and $B^\#$ can be formulated as a max-flow problem. To represent graph networks, we will use the notation (V, E, c), where V and E are the vertex and edge sets, respectively, and $c : E \to \mathbb{N}$ gives the edge capacities. Consider the network

$$G(C, \mu, A^\#, B^\#) = (\{s, t\} \cup U \cup V, E, c)$$

where C is a candidate isometry, $U = \{u_1, \ldots, u_n\}$ represents the points in $A^\#$ and $V = \{v_1, \ldots, v_m\}$ the points in $B^\#$,

$$E = \{(s, u_i) \mid 1 \le i \le n\} \cup \{(v_j, t) \mid 1 \le j \le m\} \cup \{(u_i, v_j) \mid dist(I(a_i^\#), b_j^\#) \le \mu\},$$

and $c(e) = 1$ for all $e \in E$. The max-flow on G is equal to $k_{max}(\mu, A^\#, B^\#; C)$.

At this point, we exploit the special structure of the perturbed sets $A^\#$ and $B^\#$, in order to efficiently compute a max-flow. Feder and Motwani [FM] have presented a technique for improving graph algorithms called the *compression graph*, which is well suited to $A^\#$ and $B^\#$. Let $U_k^\circ = \{u_i \mid a_i \text{ is moved onto } a_k^\circ\} \subset U$ and $V_l^\circ = \{v_j \mid b_j \text{ is moved onto } b_l^\circ\} \subset V$. If C moves gridpoint a_k° into the μ-neighborhood of b_l°, this generates a complete bipartite subgraph with node sets U_k° and V_l°. For every such bipartite clique, we remove the edges in $U_k^\circ \times V_l^\circ$, add a new node w_{kl}, and add edges $\{(u_i, w_{kl}) \mid u_i \in U_k^\circ\}$ and $\{(w_{kl}, v_j) \mid v_j \in V_l^\circ\}$ with capacity 1 each. Thereby we replace $|U_k^\circ| \cdot |V_l^\circ|$ old edges by $|U_k^\circ| + |V_l^\circ|$ new edges. We call the resulting graph $G^\#_{\text{comp}}(C, \mu, A^\#, B^\#)$.

A max-flow in $G^\#_{\text{comp}}(C, \mu, A^\#, B^\#)$ corresponds to a max-flow in $G(C, \mu, A^\#, B^\#)$. We can show that Dinic's algorithm computes a max-flow on $G^\#_{\text{comp}}(C, \mu, A^\#, B^\#)$ in $O(\sqrt{n+m})$ phases, each in time proportional to the new number of edges. The reason for this is that Dinic's algorithm takes $O(\sqrt{\text{number of nodes}})$ phases in a simple 0-1 network. A node is called simple if it has indegree 1 or outdegree 1, and a network is called simple if all nodes are simple. Note that the nodes w_{kl} are not simple in the compression of a graph; however, in a manner analogous to the proof on the number of phases of Dinic's algorithm for simple 0-1 networks in [Me, page 81], it can be shown that Dinic's algorithm takes $O(\sqrt{\text{number of simple nodes}})$ phases in a 0-1 network, if no edge connects non-simple nodes.

The μ-neighborhood of any point contains $O((\mu/\gamma)^d)$ grid points (since $\gamma = \Theta(\lambda)$). Hence each point of $A^\# \cup B^\#$ contributes to $O((\mu/\gamma)^d)$ bipartite cliques. Thus the number of edges in graph $G^\#_{\text{comp}}(C, \mu, A^\#, B^\#)$ is $O((n+m)(\mu/\gamma)^d)$, which yields a total time of $O((n+m)^{1.5}(\mu/\gamma)^d)$ to compute $k_{max}(\mu, A^\#, B^\#; C)$. The graph $G^\#_{\text{comp}}$ can be constructed in time $O((n+m)\log^{d-1}(n+m))$ using standard range-searching techniques [PS].

Note that the original graph G is a simple 0-1 network, for which a max-flow can be computed in $O((n+m)^{1/2}nm)$ time. Since $m = \Theta(n)$ for a typical problem instance, we see that the use of the compression graph $G^{\#}_{\text{comp}}$ that is enabled by the structure of the perturbed sets $A^{\#}$ and $B^{\#}$ reduces the run-time in terms of n.

We now discuss the construction of the set of candidates $\mathcal{C}$ for various isometry groups. Throughout this section, we let $\delta = \max\{\text{diam}(A), \text{diam}(B)\}$ be the larger of the diameters of the sets A and B. For each isometry group $\mathcal{I}$ we must find a set of candidates $\mathcal{C}$ that satisfies:

Condition 1: For each $I \in \mathcal{I}$ that admits (ϵ,k)-congruence for some $k \in \{2,\ldots,n\}$, there exists a candidate isometry $C \in \mathcal{C}$ such that $dist(C(a^{\#}), I(a^{\#})) \leq \gamma_1$ for all $a^{\#} \in A^{\#}$.

If $\mathcal{I}$ equals the set of translations on $\mathbb{R}^d$, then let T_{xy} represent the translation that maps point x to point y. By fixing point x and choosing as candidates all translations T_{xy} where y is a lattice point, we satisfy the above condition, with $\gamma_1 = \sqrt{d}\lambda/2$ (recall that each point of $\mathbb{R}^d$ is no more than distance $\sqrt{d}\lambda/2$ from the nearest lattice point). In order to obtain a finite set of candidates, we note that if a candidate C admits (ϵ,k)-congruence for some $k \geq 2$, then C maps the convex hulls of $A^{\#}$ and $B^{\#}$ within a distance ϵ of each other. Thus, a collection $\mathcal{C}$ of $|\mathcal{C}| = O(((\epsilon+\delta)/\lambda)^d) = O((\delta/\gamma)^d)$ candidates suffices.

Let $\mathcal{I}$ be the set of isometries on $\mathbb{R}^d$. Since an isometry is an affine mapping, it can be described by its action on $d+1$ affinely independent points. We will construct a collection $\mathcal{C}$ that satisfies Condition 1, where each candidate isometry is a set of $d+1$ affinely independent points. Fixing the first point is equivalent to choosing a translation. Once the first point is fixed, each remaining point is free to move on a d-dimensional hypersphere centered at the first fixed point, and after $d' \leq d-1$ affinely independent points have been fixed, each point not in the affine subspace of the fixed points is free to move on a $(d-d'+1)$-dimensional hypersphere. Once d affinely independent points have been fixed, each point not in the affine subspace can assume one of two possible values, since only a reflection remains to be determined.

We construct the candidates by choosing fixed points one at a time. For the first point we choose $O((\delta/\gamma)^d)$ points in such a manner that every translation that causes the convex hulls of $A^{\#}$ and $B^{\#}$ to intersect has its fixed point within distance $O(\gamma)$ of the fixed point of a candidate. When $d' \leq d-1$ points have been fixed, the set of potential choices for the next fixed point is represented by all points on a hypersphere. Since no free point moves on a hypersphere of radius greater than δ, we choose $O((\delta/\gamma)^{(d-d'+1)})$ candidate points on the hypersphere of radius δ centered at the first fixed point, such that no point on this hypersphere is more than distance $O(\gamma)$ from a candidate point. The last fixed point represents selecting a reflection and therefore presents only two choices. If the dimension d is a constant, then by choosing λ and the constant in the $O(\gamma)$ term above sufficiently small, Condition 1 is satisfied. The total number of candidates is $O((\delta/\gamma)^{d(d+1)/2})$.

6 Time complexity

The run-time of a (γ,γ)-approximate algorithm for $k_{max}(\varepsilon)$ is equal to $O(T \cdot |\mathcal{C}|)$, where $|\mathcal{C}|$ equals the number of candidates, and T the time to test each candidate of $\mathcal{C}$. We saw above that $T = O((n+m)^{1.5}(\varepsilon/\gamma)^d)$. The value $|\mathcal{C}|$ varies with the isometry family, and with the above observations we obtain the following time-complexities for a (γ,γ)-approximate algorithm for $k_{max}(\varepsilon)$: $O((n+m)^{1.5}(\varepsilon/\gamma)^d(\delta/\gamma)^d)$ if $\mathcal{I}$ is the set of translations in $\mathbb{R}^d$ for $A, B \in \mathbb{R}^d$, and $O((n+m)^{1.5}(\varepsilon/\gamma)^d(\delta/\gamma)^{d(d+1)/2})$ if $\mathcal{I}$ is the set of isometries in $\mathbb{R}^d$ for $A, B \in \mathbb{R}^d$.

Let us examine the total run-time to construct a (ε,k,ν)-approximate map. To compute the number of Phase I tests for a given k, consider that we test for (ε,k)-congruence for all values $\varepsilon = 2^i\nu$, $i = 0,\ldots,j$, where $\varepsilon_{opt}(k) \in (2^{j-1}\nu, 3\cdot 2^{j-1}\nu]$. This implies $j = \log_2(\varepsilon_{opt}(n)/\nu)$ tests until we receive a YES or MAYBE answer for all k, since $k = n$ will be the last to receive such an answer. Also, at each value $\varepsilon = 2^i\nu$, one additional test may have to be performed because certain values of k receive for the first time there an answer of MAYBE, but one test suffices even if several values of k receive a MAYBE at the same value $\varepsilon = 2^i\nu$. Thus, the total number of values of ε tested in Phase I is $O(\log(\varepsilon_{opt}(n)/\nu))$; since $\gamma = \varepsilon/2$ or $\gamma = \varepsilon/4$ at all times during Phase I, the total time is $O((n+m)^{1.5}(\delta/\nu)^x \log(\varepsilon_{opt}(n)/\nu))$, where the exponent x is determined by the family of isometries $\mathcal{I}$ and the dimension d.

Phase II, for a given k, is a binary search on an interval $(c, 2c]$ that contains $\varepsilon_{opt}(k)$. Each search query is a test for (ε,k)-congruence, where ε is in $(c, 2c]$ and thus approximately equal to $\varepsilon_{opt}(k)$, while γ is cut in half after each query. Therefore, since $T = O((n+m)^{1.5}(\varepsilon/\gamma)^d)$ and $|\mathcal{C}|$ varies inversely with γ for isometry families that we considered, the sum of all $k_{max}(\varepsilon)$ tests always will be dominated by the last one, for which γ is approximately the same size as ν. If we perform Phase II over all n values of k, we get a total time of $O(n(n+m)^{1.5}(\delta/\nu)^x(\varepsilon_{opt}(n)/\nu)^d)$, with x as above, since $\varepsilon_{opt}(n) \geq \varepsilon$ for every test value ε. We see that Phase II dominates Phase I.

In summary, implementing Phase I and Phase II together builds a (ε,k,ν)-approximate map. Using the (γ,γ)-approximate algorithms for $k_{max}(\varepsilon)$ that we have described yields the following time-bounds for constructing (ε,k,ν)-approximate maps:

$O(n(n+m)^{1.5}(\varepsilon_{opt}(n)/\nu)^d(\delta/\nu)^d)$ if $\mathcal{I}$ is the set of translations in $\mathbb{R}^d$, for $A, B \in \mathbb{R}^d$;

$O(n(n+m)^{1.5}(\varepsilon_{opt}(n)/\nu)^d(\delta/\nu)^{d(d+1)/2})$ if $\mathcal{I}$ is all isometries in $\mathbb{R}^d$, for $A, B \in \mathbb{R}^d$.

7 Projective congruence

In *projective congruence*, we compare a 2-dimensional point set B to a 3-dimensional set A. Rather than choosing just an isometry and a matching (between the elements of the point sets), finding a congruence between the points sets now consists of first choosing an orthogonal projection of the 3-dimensional set A to $\mathbb{R}^2$, and then choosing an isometry on the resulting 2-dimensional set, and finally a matching. We develop an approximate algorithm for this problem by approaching it in a similar manner as before. The Phase I-Phase II approach

motivates a version of the $k_{max}(\varepsilon)$ problem, in which the set of all possible compositions of projection and isometry on A must be discretized into candidates that closely approximate all reasonable choices. Discretizing the candidates resembles the same task for isometries in $\mathbb{R}^3$, except that the projection of the point set to an affine space of one lesser degree results in the loss of one degree of freedom; in fixing the first point (which determines the translation), we realize that we have two dimensions of freedom and not three, since motion parallel to the direction of projection is irrelevant. The total number of candidates on the last test of Phase II is $O((\delta/\nu)^5)$, and the max-flow problems here are between point sets in $\mathbb{R}^2$, so the total run-time is $O(n(n+m)^{1.5}(\varepsilon_{opt}(n)/\nu)^2(\delta/\nu)^5)$.

8 Conclusion

We have generalized the concept of ε-congruence in order to encompass a greater number of situations; our generalized definition is called (ε,k)-congruence, and it is represented pictorially by the (ε,k)-map. Because we desired small run-times, we chose to develop approximate algorithms that test for (ε,k)-congruence, using many of the techniques of [HS]. While we did not discuss complete algorithms for (ε,k)-congruence, such algorithms would extend naturally from some of the complete algorithms for ε-congruence given by [AMWW].

While we have achieved run-times with small exponents for n and m, it is troubling that the diameter δ appears. This suggests that our approximate al orithms are best suited for relatively dense point sets. Ironically, for ε-congruence, complete algorithms perform best on point sets that are *ε-separated* [AMWW, AKMSW] (i.e. no two points of the same set are within distance ε of each other). One open question is to explore which types of point sets are best suited to our approximate algorithms and which to complete algorithms. As a final open question, we ask if there are ways to reduce $|\mathcal{C}|$ for the various isometry families we studied, since any reduction leads automatically to improved algorithms for $k_{max}(\varepsilon)$ and the (ε,k,ν)-approximate map.

References

[AMWW] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl, "Congruence, Similarity, and Symmetries of Geometric Objects", *Discrete and Computational Geometry*, **3** (1988), pp. 237-256.

[AKMSW] E.M. Arkin, K. Kedem, J.S.B. Mitchell, J. Sprinzak, M. Werman, "Matching Points into Noise Regions: Combinatorial Bounds and Algorithms", *ORSA J. on Computing*, 4 (1992), pp. 375-386.

[Ba] H.S. Baird, *Model-Based Image Matching Using Location*, MIT Press, 1984.

[FM] T. Feder and R. Motwani, "Clique Partitions, Graph Compression and Speeding-up Algorithms", in *Proc. of the 23rd ACM Symp. on Theory of Computing*, 1991.

[HS] P.J. Heffernan and S. Schirra, "Approximate Decision Algorithms for Point Set Congruence", in *Proc. of 8th ACM Symp. on Computational Geometry* (1992), pp. 93-101.

[Ma] G.E. Martin, *Transformation Geometry*, Springer Verlag, 1982.

[Me] K. Mehlhorn, *Data Structures and Algorithms 2: Graph Algorithms and NP-completeness*, Springer-Verlag, 1984.

[PS] F.P. Preparata and M.I. Shamos, *Computational Geometry — An Introduction*, Springer-Verlag, 1985.

[Sc1] S. Schirra, *Über die Bitkomplexität der ε-Kongruenz*, Diplomarbeit, Universität des Saarlandes, Saarbrücken, Germany, 1988.

[Sc2] S. Schirra, "Approximate Decision Algorithms for Approximate Congruence", *Information Processing Letters*, **43** (1992), pp. 29-34.